

```

RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSSSSSS
RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSSSSSS
RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSSSSSS
RRR      RRR      MMMMMM      MMMMMM      SSS
RRR      RRR      MMMMMM      MMMMMM      SSS
RRR      RRR      MMMMMM      MMMMMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSS
RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSS
RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      SSSSSSSSSSSSSS
RRR      RRR      MMM      MMM      SSSSSSSSSSSSSS
RRR      RRR      MMM      MMM      SSSSSSSSSSSSSS

```

53

Syn

NTS

NTS
NTS

NTS

NTS
NTS

N13

NTS

NTS
NTS

NTS

NTS
NTSNTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTSNTS
NTS

NTS

NTS

NTA

NTS
NTSNTS
NTS

NTS

NTS
NTS

NTS

10

10

NTC

NT
NT

NTS

NT
NT

NT

PI

10

100

100

10

10

11. *Journal of the American Medical Association*, 273, 1995, 1033-1034.

1

```
RRRRRRRR  MM      MM      11      WW      WW  TTTTTTTTTT  LL      SSSSSSSS  TTTTTTTTTT
RRRRRRRR  MM      MM      11      WW      WW  TTTTTTTTTT  LL      SSSSSSSS  TTTTTTTTTT
RR      RR  MMMM  MMMM  1111  WW      WW      TT      LL      SS      TT
RR      RR  MMMM  MMMM  1111  WW      WW      TT      LL      SS      TT
RR      RR  MM  MM  MM      11  WW      WW      TT      LL      SS      TT
RR      RR  MM  MM  MM      11  WW      WW      TT      LL      SS      TT
RRRRRRRR  MM      MM      11  WW      WW      TT      LL      SSSSSS  TT
RRRRRRRR  MM      MM      11  WW      WW      TT      LL      SSSSSS  TT
RR      RR  MM      MM      11  WW      WW      TT      LL      SS      TT
RR      RR  MM      MM      11  WW      WW      TT      LL      SS      TT
RR      RR  MM      MM      11  WWW  WWW  TT      LL      SS      TT
RR      RR  MM      MM      11  WWW  WWW  TT      LL      SS      TT
RR      RR  MM      MM      11  WWW  WWW  TT      LL      SS      TT
RR      RR  MM      MM      11  WWW  WWW  TT      LL      SS      TT
RR      RR  MM      MM      111111  WW      WW      TT      LLLLLLLLLL  SSSSSSSS  TT
RR      RR  MM      MM      111111  WW      WW      TT      LLLLLLLLLL  SSSSSSSS  TT
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```


(2) 69
(3) 95

DECLARATIONS
RMSWTLST1 - WRITE LAST BLOCK, PADDING IF AT EOF

```
0000 1          $BEGIN RM1WTLST,000,RMSRMS1,<WRITE LAST BLOCK SEQ.>
0000 2
0000 3
0000 4 :*****
0000 5 :*
0000 6 :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 :*  ALL RIGHTS RESERVED.
0000 9 :*
0000 10 :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 :*  TRANSFERRED.
0000 16 :*
0000 17 :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 :*  CORPORATION.
0000 20 :*
0000 21 :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 :*
0000 24 :*
0000 25 :*****
0000 26 :
0000 27 :++
0000 28 : Facility: rms32
0000 29 :
0000 30 : Abstract:
0000 31 :
0000 32 :   this module writes out the last block
0000 33 :   of a sequential file if at eof and dirty.
0000 34 :
0000 35 : Environment:
0000 36 :   star processor running starlet exec.
0000 37 :
0000 38 : Author: l f laverdure,      creation date: 29-march-77
0000 39 :
0000 40 : Modified By:
0000 41 :
0000 42 :   V03-003 MCN0001      Maria del C. Nasr      30-Nov-1982
0000 43 :   Last block should be at least 14 bytes long for foreign
0000 44 :   magtape, not 18 bytes.
0000 45 :
0000 46 :   V03-002 KBT0151      Keith B. Thompson      20-Aug-1982
0000 47 :   Reorganize psects
0000 48 :
0000 49 :   V03-001 JWH0002      Jeffrey W. Horn 2-Jun-1982
0000 50 :   On writing short block to foriegn mag-tape, don't
0000 51 :   round-up byte count.
0000 52 :
0000 53 :   V02-013 JWH0001      Jeffrey W. Horn 20-Jan-1982
0000 54 :   On flush clear IFB$W_FFB if magtape.
0000 55 :
0000 56 :   V02-012 RAS0020      Ron Schaefer      6-Aug-1981
0000 57 :   Correct minimum buffer size for magtape.
```


RM1WTLST
V04-000

WRITE LAST BLOCK SEQ.

N 15

16-SEP-1984 00:59:26
5-SEP-1984 16:23:57

VAX/VMS Macro V04-00
[RMS.SRC]RM1WTLST.MAR;1

Page 2
(1)

0000 58 :
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
0000 64 :
0000 65 :--
0000 66 :
0000 67 :

V02-011 RAS0016 Ron Schaefer
Add stream file support.

6-Aug-1981

V02-010 REFORMAT Ron Schaefer
Reformat the source.

30-Jul-1980

09:13


```
0000 69      .SBTTL  DECLARATIONS
0000 70
0000 71 ::
0000 72 :: Include Files:
0000 73 ::
0000 74
0000 75 ::
0000 76 :: Macros:
0000 77 ::
0000 78
0000 79      $IFBDEF
0000 80      $IRBDEF
0000 81      $BDBDEF
0000 82      $FABDEF
0000 83      $RABDEF
0000 84      $DEVDEF
0000 85
0000 86 ::
0000 87 :: Equated Symbols:
0000 88 ::
0000 89
0000 90 ::
0000 91 :: Own Storage:
0000 92 ::
0000 93
```



```

0000 95      .SBTTL  RM$WTLST1 - WRITE LAST BLOCK, PADDING IF AT EOF
0000 96
0000 97      :++
0000 98      RM$WTLST1:  write last block, padding if at eof
0000 99
0000 100     Calling sequence:
0000 101
0000 102     bsbw      rm$wtlst1
0000 103
0000 104     Input Parameters:
0000 105
0000 106     r11      impure area address
0000 107     r10      ifab address
0000 108     r9       irab address
0000 109     r8       rab address
0000 110
0000 111     Implicit Inputs:
0000 112
0000 113     the contents of the various internal structures,
0000 114     in particular:
0000 115
0000 116     irb$l_curbdb
0000 117     irb$v_eof
0000 118     irb$w_nrp_off
0000 119     bdb$v_drt
0000 120     bdb$w_numb
0000 121     ifb$b_rfm
0000 122     ifb$v_ansi_d
0000 123     ifb$l_prim_dev
0000 124     the various inputs to rm$relblk1 and rm$mapblk1
0000 125
0000 126     Output Parameters:
0000 127
0000 128     r0          status code
0000 129     r1-r4,ap destroyed
0000 130
0000 131     Implicit Outputs:
0000 132
0000 133     the current bdb is released, with
0000 134     padding and writing if necessary.
0000 135
0000 136     Completion Codes:
0000 137
0000 138     standard rms, as per rm$relblk1.
0000 139
0000 140     Side Effects:
0000 141
0000 142     may have switched to running at ast level.
0000 143
0000 144     --
0000 145
0000 146     RM$WTLST1::
0000 147     $STPT  WTLST1
0000 148     MOVL   IRB$L_CURBDB(R9),R4      ; get current bdb
0000 149     BEQL   WTLXIT                    ; branch if none
0000 150     BBC    #IRB$V_EOF,(R9),REL1    ; branch if not eof
0000 151     BBC    #BDB$V_DRT,BDB$B_FLGS(R4),REL1 ; or if buffer not dirty

```

54	20	A9	D0	0006	148		
		69	13	000A	149		
51	69	21	E1	000C	150		
4C	0A	A4	01	E1	0010	151	


```
0015 152
0015 153 ;
0015 154 ; pad out last block if not full or make
0015 155 ; it a short write if appropriate
0015 156 ; (note: unit record device will not be dirty, hence won't come here).
0015 157 ;
0015 158
57 DD 0015 159 PUSHL R7
FFE6' 30 0017 160 BSBW RMS$MAPBLK1 ; map the buffer
001A 161 ; r1 = buffer addr
001A 162 ; r7 = buffer end addr+1
52 44 A9 3C 001A 163 MOVZWL IRB$W_NRP_OFF(R9),R2 ; get offset in block
04 12 001E 164 BNEQ 10$ ; branch if non-zero
3A 6A 05 E1 0020 165 BBC #DEV$V_SQD,IFB$S_PRIM_DEV(R10),RELEASE ; branch if not magtape
0024 166
0024 167 ;
0024 168 ;!!!!!!
0024 169
0024 170 NOTE:
0024 171 The above test causes a magtape that has been truncated (the only case
0024 172 that could leave a zero length buffer dirty) to have a block written
0024 173 back out so that the truncate really takes effect. For variable length
0024 174 records an entire block of pad characters is written. For fixed length
0024 175 records an attempt is made to write a zero length block, which eventually
0024 176 fails. This is a legitimate bug, which should be fixed in truncate by
0024 177 telling the magtape acp that the function is truncate, not backspace,
0024 178 if and when the acp gets the function. When this occurs the above code
0024 179 to special case for magtape can be reworked.
0024 180
0024 181 ;!!!!!!
0024 182 ;
0024 183
50 51 52 C0 0024 184 10$: ADDL2 R2,R1 ; get addr of next byte
57 51 C3 0027 185 SUBL3 R1,R7,R0 ; get # bytes left
FFD2' 30 002B 186 BSBW RMS$PADBLK1 ; pad out the block
2C 6A 05 E1 002E 187 BBC #DEV$V_SQD,IFB$S_PRIM_DEV(R10), -
0032 188 RELEASE ; branch if not magtape
5C AA B4 0032 189 CLRW IFB$W_FFB(R10) ; clear first free byte field
01 50 AA 91 0035 190 CMPB IFB$B_RFMORG(R10),#FAB$C_FIX ; fixed length records?
23 12 0039 191 BNEQ RELEASE
003B 192
003B 193 ;
003B 194 ; fixed length records.
003B 195 ; if tape, write a short block. (at least 18 bytes, 14 for foreign tapes)
003B 196 ;
003B 197
52 44 A9 3C 003B 198 MOVZWL IRB$W_NRP_OFF(R9),R2 ; get # of bytes
OF 6A 18 E0 003F 199 BBS #DEV$V_FOR,IFB$S_PRIM_DEV(R10),20$ ; branch if foreign
52 52 D6 0043 200 INCL R2 ; round up if not
12 01 AA 0045 201 BICW2 #1,R2
52 52 B1 0048 202 CMPW R2,#18 ; buffer big enough?
0D 1E 004B 203 BGEQU 30$ ; OK
52 12 B0 004D 204 MOVW #18,R2 ; at least 18 (already padded)
08 11 0050 205 BRB 30$ ; skip foreign tapes
0E 52 B1 0052 206 20$: CMPW R2,#14 ; buffer big enough for foreign tape?
03 1E 0055 207 BGEQU 30$ ; OK
52 0E B0 0057 208 MOVW #14,R2 ; at least 14 (already padded)
```



```
14 A4 52 B0 005A 209 30$: MOVW R2,BDB$W_NUMB(R4) ; and set length of buffer
          005E 210
          005E 211 ;
          005E 212 ; release buffer, writing and extending if needed
          005E 213 ;
          005E 214
          005E 215 RELEASE:
57 8E DO 005E 216 REL1: MOVL (SP)+,R7
FF9C' 30 0061 217 REL1: BSBW RMS$RELBLK1
OE 50 E8 0064 218 BLBS R0,WTLXIT ; branch if all ok
50 DD 0067 219 PUSHL R0 ; save status
OC A8 DD 0069 220 PUSHL RAB$L_STV(R8)
07 10 006C 221 BSBB WTLXIT ; quiet all other io if any
OC A8 8E DO 006E 222 MOVL (SP)+,RAB$L_STV(R8) ; restore status
01 BA 0072 223 POPR #^M<R0>
05 0074 224 RSB
          0075 225
          0075 226 ;
          0075 227 ; quiet all rah/wbh io if any and exit
          0075 228 ;
          0075 229
FF88' 31 0075 230 WTLXIT: BRW RMS$QUIET_SEQMBF ; make sure all io is done and return
          0078 231
          0078 232 .END
```


RM1WTLST
Symbol table

WRITE LAST BLOCK SEQ.

F 16

16-SEP-1984 00:59:26 VAX/VMS Macro V04-00
5-SEP-1984 16:23:57 [RMS.SRC]RM1WTLST.MAR;1

Page 7
(3)

\$\$PSECT_EP	=	00000000		
\$\$RMSTEST	=	0000001A		
\$\$RMS_PBUGCHK	=	00000010		
\$\$RMS_TBUGCHK	=	00000008		
\$\$RMS_UMODE	=	00000004		
BDB\$B_FLGS	=	0000000A		
BDB\$V_DRT	=	00000001		
BDB\$W_NUMB	=	00000014		
DEV\$V_FOR	=	00000018		
DEV\$V_SQD	=	00000005		
FAB\$C_FIX	=	00000001		
IFB\$B_RFMORG	=	00000050		
IFB\$B_PRIM_DEV	=	00000000		
IFB\$W_FFB	=	0000005C		
IRB\$B_CURBDB	=	00000020		
IRB\$V_EOF	=	00000021		
IRB\$W_NRP_OFF	=	00000044		
PIOSA_TRACE	*****	X	01	
RAB\$B_STV	=	0000000C		
REL1	00000061	R	01	
RELEASE	0000005E	R	01	
RMS\$MAPBLK1	*****	X	01	
RMS\$PADBLK1	*****	X	01	
RMS\$QUIET_SEQMBF	*****	X	01	
RMS\$RELBLK1	*****	X	01	
RMS\$WTLST1	00000000	RG	01	
TPT\$B_WTLST1	*****	X	01	
WTLXIT	00000075	R	01	

+-----+
! Psect synopsis !
+-----+

PSECT name

PSECT name	Allocation	PSECT No.	Attributes															
. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE					
RMS\$RMS1	00000078 (120.)	01 (1.)	PIC	USR	CON	REL	GBL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE					
\$ABSS	00000000 (0.)	02 (2.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.07	00:00:01.01
Command processing	125	00:00:00.76	00:00:07.04
Pass 1	256	00:00:06.99	00:00:16.94
Symbol table sort	0	00:00:00.99	00:00:01.23
Pass 2	63	00:00:01.36	00:00:50.19
Symbol table output	3	00:00:00.06	00:00:00.40
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	482	00:00:10.25	00:01:16.83

The working set limit was 1350 pages.
40127 bytes (79 pages) of virtual memory were used to buffer the intermediate code.

There were 40 pages of symbol table space allocated to hold 800 non-local and 3 local symbols.
232 source lines were read in Pass 1, producing 13 object records in Pass 2.
16 pages of virtual memory were used to define 15 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
-----	-----
\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	7
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	11

886 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RM1WTLST/OBJ=OBJ\$:RM1WTLST MSRC\$:RM1WTLST/UPDATE=(ENH\$:RM1WTLST)+EXECML\$/LIB+LIB\$:RMS/LIB

0322

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY